

# An Approach for Developing Applications in Lotus Notes

**ABSTRACT:** Lotus Notes, an innovative groupware, is evolving from its origins of facilitating collaboration among small groups into an integrated and open platform for building and running client-server applications for automating workflow and office activities. Since both the application domain and the development environment are new, the development process for Lotus Notes applications is naturally ad hoc. To facilitate the creation of Notes applications, Lotus Development Corporation has provided a set of templates that can be configured and adapted by the application developers. However, the issues of how to analyze the applications domain so that the requirements can be mapped on one or more of these templates has not been addressed in the literature. The analysis problem becomes more acute because the key attributes of the applications domain and the Notes environment, namely, the collaboration among physically dispersed users who have different access rights to data that resides on distributed databases, are significantly different from MIS/database applications and the traditional analysis methods are not directly applicable. This paper presents and illustrates an orderly approach for developing applications in Lotus Notes that addresses the issues of analyzing the problem domain so that the requirements can be mapped onto a Lotus Notes template.

**Keywords:** *Groupware, Lotus Notes, Applications Development, Client/Server, Workflow*

## INTRODUCTION

Lotus Notes is an innovative groupware product of Lotus Development Corporation. Physically, Notes is an environment where personal computer users over local- and wide-area networks communicate information and share databases to perform their portions of the application tasks. This computer-based work environment where group members, who may be geographically dispersed, communicate and share information without any predefined order or time represents a new computing paradigm, called Computer-Supported Cooperative Work, or CSCW. Today the more user-friendly and marketable term, Groupware, has virtually replaced CSCW.

Groupware applications are quite different from the traditional management information systems (MIS) applications, and they need new development strategies and methods. Specifically, it has been found (7, 8) that group work is hard to describe from an information flow perspective and the traditional systems analysis and design methodologies are deficient for modeling groupware applications. It should also be noted that in the Lotus Notes environment end-users invariably become applications developers. Although this notion of end-users becoming applications developers is not new because the penetration of personal

computers in the work place created both the opportunity and need for end-users to build their own applications, the scope of the problem being addressed has dramatically changed. For example, in a typical personal computer environment most end-user built applications are created for personal use. In Lotus Notes, applications are always built for group use where the group members often cross departmental and organizational boundaries. This change in the number of users of an application from one to many alters the characteristics of the application in terms of both the quality and behavior of the requirements of the application. This is the reason why MIS applications are generally developed by information systems professionals (IS) who use formal methodologies and computer tools to systematically address the complexities of multi-user applications.

To ease the task of developing Notes applications, Lotus Development Corporation provides a set of templates that can be configured and adapted to build user applications. Experience has shown that it is not easy to correctly adapt a template to build a workflow or office automation application. There are two reasons for this. First, the Notes implementation environment is very rich and the development language is functional. Second, the traditional MIS analysis

**Donald R. Chand**

Computer Information Systems Department  
Bentley College  
Waltham, MA 02154  
1-617- 891-2925  
dchand@bentley.edu

methods are inadequate for analyzing and generating the requirements for this new domain of workflow and office activities. This paper presents a pragmatic approach that can be used by both end-users and IS professionals to systematically build Lotus Notes applications. Specifically, the paper describes and illustrates a systematic approach for analyzing a Lotus Notes application in such a way that the requirements can be directly mapped into the structure and elements of a Notes template. This approach is different from the design philosophy of Winograd (10) who suggests that group work needs to be modeled using a language-action perspective.

This paper begins with a discussion of the evolution of groupware computing paradigm whose objective is to expose the reader to the computing facilities embodied in Lotus Notes. Next, a functional architecture of the Lotus Notes is sketched to provide a conceptual model of the Notes environment. The fourth section describes and illustrates the analysis and design steps for developing Lotus Notes applications. The fifth section presents a systematic approach for analyzing a Lotus Notes template. The sixth section shows how to link the application requirements to the template elements and complete the development of an application.

## THE ORIGIN OF GROUPWARE

During the 1980s, we witnessed the invasion of microcomputers to support office work. Word processing software on these microcomputers replaced the typewriter. Professionals who needed to do repeated recalculations found that spreadsheet packages saved enormous computational effort and also opened a way to perform deeper and more comprehensive "what if" analyses. Database software packages on these personal computers provided the capability to expedite the use and search of data lists needed to support numerous business tasks. These PC tools amplified the single user productivity but failed to do for office work what automation did for factory work. The reason for this is that automating individualistic problem-solving tasks like number crunching, writing, and the preparation of eye-catching presentations do not address the central tasks of office work — which is mainly about coordinating, communicating, collaborating, and negotiating with groups who are increasingly distributed globally.

The essence of group work is communications and sharing. Groups spend a significant portion of their time in communicating, sharing, tracking, and managing tasks and information associated with their work and projects. Until the availability of computer networks and groupware, the communications and sharing was supported by conventional channels, such as phone conversation, document distribution and various mailing facilities.

Workflow automation has been evolving over the years. The initial attempt was to automate the hard copy forms. This aided in record maintenance and revisions. The next step was process automation, where the flow of documents was defined and charted and an automated system was developed to move these documents along the established process. This streamlined process expedited the flow. However, the inflexibility of the predefined process limited its use in meeting the somewhat unpredictable nature of information sharing and communication among the workers. Eventually it was recognized that significant productivity gains will not occur until the user has full freedom to decide when to communicate and exchange information and also has control over the type of information to communicate.

The software that embraces this concept of group communication and sharing is called groupware. But what is groupware? Bullen and Bennett (1) define groupware as

Figure 1. CONCEPTUAL ARCHITECTURE OF NOTES DEVELOPMENT ENVIRONMENT



computer-based tools that can be used by work groups to facilitate the exchange and sharing of information. Ellis and Rein (3) define it as computer-based systems that support two or more users engaged in a common task by providing an interface to a shared environment. These characterizations of groupware are very broad. As a result, enabling technologies like standard electronic mail capability with few bells and whistles or a routing package that sits on the email backbone and allows groups to share project documents are classified as groupware. A package like Lotus Notes, which offers a forms based application development environment and database access on top of email and routing facilities is providing a benchmark. Therefore, to gain an understanding of groupware, we outline the fundamental features of Notes. Lotus Notes supports:

- Email
- Object linking & Embedding
- Conferencing
- Document Routing
- Database Access
- Shared Filing
- Tracking
- Document Libraries

Email is the most common communicating mechanism among users. It moves files on the network and allows users to forward, reply, use a distribution list to send messages, and provides text editing and file export / import capabilities. Object Linking and Embedding is a means to provide a seamless environment to share work such as word-processing documents, spreadsheet models and graphics outputs, over networks

from the program in which it was created. Conferencing is an environment that enables work-group members to share information, ask questions and offer opinions on various subjects irrespective of geographical distances. Document Routing is a facility that enables users to route documents for review or approval on a specific routing order. The document routing module will be able to furnish the status of current documents. Database access allows users to browse, view, create, update and extract data from the primary database to perform their portion of the over all task or work. Shared Filing means efficient sharing of real-time data and documents within and between work groups. Members contribute to these shared documents by creating new documents or responding to existing documents. Tracking means allowing team members to track project activity, monitor open issues, check on current status, answer questions, delegate assignments, and identify and remove bottlenecks. Document Libraries is a repository of reference materials and up-to-date information from newswire information feeds. Users can search the repository using keywords. In summary, Lotus Notes is an integration of electronic mail, routing, document management, and database sharing.

## ARCHITECTURE OF LOTUS NOTES DEVELOPMENT ENVIRONMENT

At the highest level of abstraction, Lotus Notes is a distributed documents management system that runs on top of an electronic messaging system, as shown in Figure 1. The electronic messaging system provides a seamless platform to microcomputer users to exchange, distribute and route their

Figure 2. DOCUMENTS MANAGEMENT SYSTEM

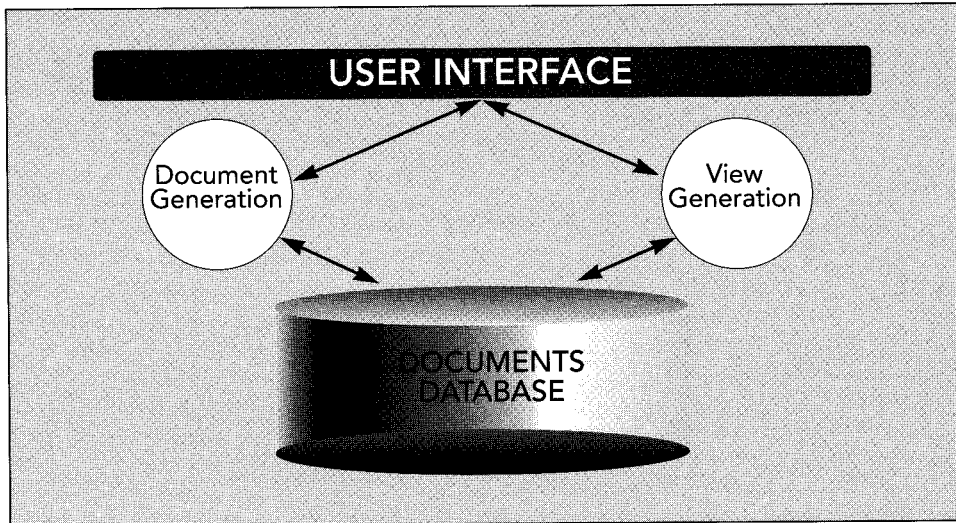
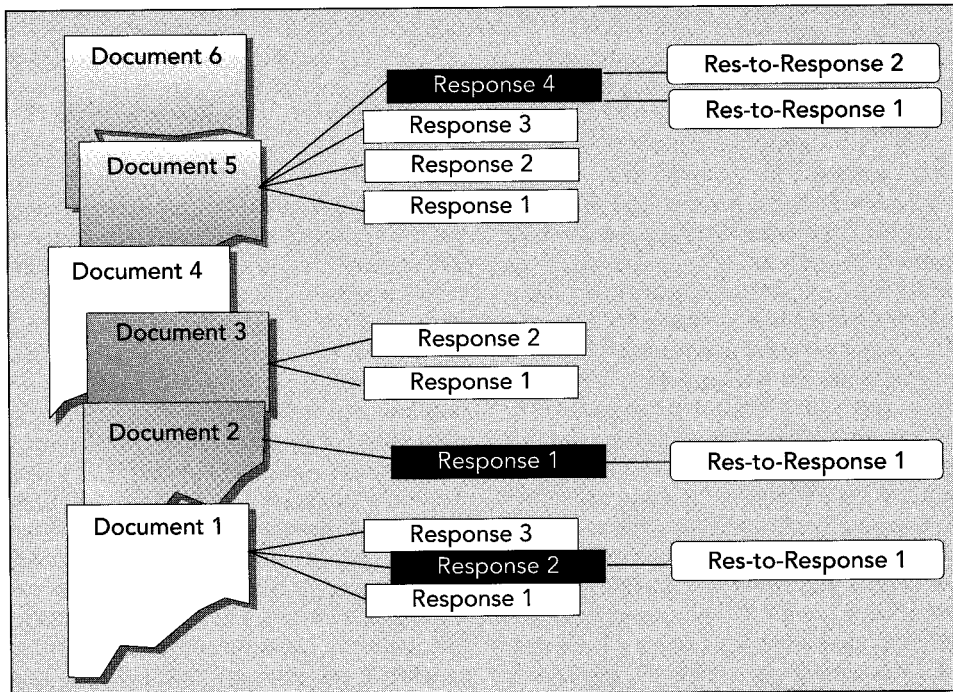


Figure 3. STRUCTURE OF NOTES DOCUMENTS DATABASE



word-processing, spreadsheet or graphics documents. The documents management system provides a forms-based graphical user interface to populate documents databases that can be shared among geographically dispersed certified users on one or more networks. The documents database and a relatively comprehensive applications development capability separates Lotus Notes from other groupware packages. For example, the primary focus of Digital Corporation's Teamlinks is on formalizing

the process of routing documents among users. It provides a rudimentary document generation facility to build a form-like document that can be routed through a specified set of users. These users can check the status of the document in terms of how far the document has traveled along the specified route. In addition, the software provides the ability to hide parts of the documents from specific users. Since the focus of Teamlinks is not on data sharing, the routing documents are deposited in each

user file cabinet for action and are not collected in a common document database as in Lotus Notes.

Notes has integrated the mailing, routing, and document creating capabilities using a common form-based user interface. In Notes the mail is generated by a mail form, and unlike other documents that are added to a shared database, a mail document is mailed to a specific user or routed through a set of users. Notes supports multiple servers over separate local area networks, provides facilities to replicate databases among multiple servers, allows remote user access to any server, and has a rich layered approach to data security for both users and servers. These facilities makes Lotus Notes a unique tool to design and run workflow applications.

Conceptually, the documents management system of Lotus Notes consists of a documents database, a user interface screen, and software modules to create documents and generate views. These elements and their interactions are shown in Figure 2. The user interface screen allows a user to compose a document by displaying the form labels, providing input fields, and prompting to enter the appropriate information. As each document is composed it is added to the central documents database.

The basic structure of Notes documents database is shown in Figure 3. The documents database consists of main form documents, responses to the main form documents, and responses to the responses. This classification of documents in terms of the main form, the response form and response to the response form provides order to group discussions. The set of documents in the databases can be tabularized in various ways that are called views. In Notes, access rights can be allocated at the database level, form level, view level and field level.

Lotus Notes connects a group of geographically dispersed users to their common application database using a star network architecture. The user on a specific server communicates with the application database on another server through the hub server. The access control list of the hub server contains the group list of users who have access rights to a specific applications database residing on a particular server. Through appropriate connection records, users on different LANs (or Notes domains) are provided access to the common applications database.

In summary, the development environment of Lotus Notes is its distributed document management system layered on top of an electronic messaging system. The document management system provides facilities to define forms, create documents and build indexes to documents for creating views. The database architecture of main forms, response forms and response to response forms is a powerful abstraction for modeling group discussion which is the central issue in groupware applications.

### AN APPROACH FOR ANALYZING LOTUS NOTES APPLICATIONS

Since communicating information and accessing shared data among geographically dispersed group members is the central issue in groupware applications, analyzing participants roles, their data access needs and the geographic locations of the users are the crucial tasks in the analysis of Lotus Notes applications. Before presenting a systematic approach for analyzing Lotus Notes applications, we will discuss the concept of "access rights" to shared data and how it differs from the traditional notion of data security.

#### The Notion of Data Access Rights

By data access rights we mean the user has the right to know the existence of information and whether he or she can create, read, or modify it. This is similar to the notion of data security in a database environment. However, the focus of data security is on protecting and limiting access, and the term is commonly used for mechanisms that protect an organization's data from common threats such as unauthorized users access, physical theft of stored data, fire and natural or unnatural disasters, physical media and drive failures, etc.. In a network environment, there is the extra line-tapping threat. In contrast, the focus of data access rights is on defining the rules for sharing data in terms of whether the user can read, create, modify or delete data.

The following scenario illustrates the data access issues in group work. Suppose the Computer Center at a College decides to automate its hot line support operations where its customers (e.g., faculty, staff and students) remotely enter their systems problems in a hot line database. The objective is that all three divisions of the Computer Center (i.e., Systems, Telecommunications, Academic Services) access this hot line database and promptly respond to the customer problem if it falls within their domain.

In this scenario there are two types of users of the hot line database: customers and computer center staff. The customers need access rights to the hot line database to log their problem. This is essentially the "minimal" rights to the database, and since it merely allows the customers to enter new data, we classify this as "depositor" right. Suppose that in this application the computer center maintains in the hot line database the status of how the problem is being addressed. If we allow the customers to monitor the progress being made on their problem, they need access rights to read the status of their problem. In a typical database environment, this access to read the status of their problem will be implemented by allocating to the customers read-only right to the hot line database. This is not an appropriate solution in our example because each customer can now browse through the problems of other customers. The need here is for the customer to be able to read the status of only his or her own problem.

Let us analyze the access needs of the computer center staff. Clearly, the computer center staff need access rights to the hot line database to read all the entries made by the customers. They also need access right to update the status of the problem. However, this update right must not allow the computer center staff the ability to modify or delete a customer entry.

The correct implementation of these access rights is crucial. The main form and response form architecture of Lotus Notes database together with the depositor, reader, author, editor, designer and management level access rights defined in Lotus Notes provide the abstractions needed to correctly model these access needs of the users. Therefore we have adapted these abstractions of Lotus Notes in our analysis process for structuring the databases and specifying appropriate access rights to these databases.

#### The Analysis Approach

The objective of the analysis approach is to organize the analysis process and tasks so that the application requirements are generated in a form that can be directly implemented in Lotus Notes. Our analysis approach consists of ten distinct steps to systematically gather and record information on the participants, their geographic locations, and the data they will share. The following groupware application is used to illustrate the analysis approach.

A professor is offering a leading edge course on advances in applications devel-

opment technologies. The students are senior systems development managers at various companies. These students are geographically distributed over the east coast and the west coast of the United States. The professor will use Lotus Notes to run this course. The course pedagogy involves: (1) the professor developing a list of topics; (2) students reading designated materials on each topic and engaging in a discussion with the professors and the other students; and (3) the professor assessing the level of participation and the quality of student input to assign a grade to each student.

**Step 1: Identify the participants and classify them into distinct categories.**

The first step in the analysis of a groupware application is to identify and classify the users into distinct categories. In our example, the participants are the professor and the students. Since the professor and the students play different roles in this application, they are placed in different categories. We create a group called PROFESSOR to represent the professor. All students are placed in another group, called STUDENT, because in this application there is no good reason to differentiate one student from another.

**Step 2: Specify any hierarchical relationship that may exist among the participants categories.**

Since the students report to the professor, a hierarchical relation can be defined between the STUDENT category and the PROFESSOR category. It is useful to capture such relations graphically, as shown below (step 8). The hierarchical relations are used for partitioning the database into main form and response form documents.

**Step 3: Specify the role of the participant in the creation and use of the common database.**

The professor is the author of a set of topics and she controls the discussion on each topic. Although the professor is the manager of this course, we cannot give the professor the manager access to the database because manager access has the right to delete students work. Therefore, as a starting point it is more appropriate to assign the professor the rights of an author only. Students use these questions to study the topic and present their understanding of the topic. They are essentially depositors of information. However, they also need read only access to the inputs of their colleagues. We found that the access categories of Lotus Notes is a good model

for specifying the access rights for the participants during the analysis phase.

- No Access—knows the database exists but cannot access it
- Depositor—can only create documents but cannot read other documents
- Reader—can read documents but can not create, modify or delete documents
- Author—can create and read documents, but cannot modify except their own documents
- Editor—can create, read and modify documents
- Designer—can create, read and modify documents and can also alter database structures and documents
- Manager—can create, read and modify documents, can alter database structures and documents, can alter user rights, and can also delete the database.

**Step 4:** Model the geographical locations of the participants into distinct networks, where a network is a cluster of users with a common server.

This information is used for formulating a database replication policy. In our example scenario the users are partitioned into two networks, the east-coast network and the west-coast network.

**Step 5:** Identify the type of information that will be shared.

The type of information that the professor will share with the students is the list of topics and the associated questions. The type of information that the students will share with the professor and with other students is their input (response) on these topics. The student input may lead the professor to generate new topic areas.

**Step 6:** Define the type of information to be shared in terms of specific data items.

In our example scenario, the data items needed to represent the type of information that the professor will share with the students are:

- ProfessorName
- DateComposed
- TopicName
- TopicCategories
- TopicQuestions

We can similarly postulate the data items needed to model students input on a topic.

- StudentName
- TodaysDate
- NewTopicName
- OriginalTopicName
- AnswerDetails

**Step 7:** For each of the data items identified in step 6, develop their full description by asking.

- What is an appropriate field name for this data item?
- What is its data type (i.e. what type of values can it take)?
- Who creates it?
- Who can modify it?
- Who can read it?
- What field security should be assigned?
- What is the entry format of the input value?
- What is the display format of the input value?
- What are the input data validation rules?
- What is the default value?
- What data entry prompts or help should be made available?

**Step 8:** Create a "data items access needs" table where data items with identical access needs are clustered together.

The clustering of data items helps in partitioning the shared data items into different forms, so that the access rights can be applied at the form level. In our example, the first five data items can be placed on the main form and the last five on the response form.

<u>DATA ITEM</u>	<u>PROFESSOR</u>	<u>STUDENT</u>
Professor Name	author	reader
DateComposed	author	reader
Topic	author	reader
TopicCategories	author	reader
TopicQuestions	author	reader
StudentName	reader	author
TodaysDate	reader	author
NewTopic	reader	author
OriginalTopic	reader	author
AnswerDetails	reader	author

**Step 9:** Define the database views that are helpful and meaningful to the participants.

In our example, both the professor and the students are interested in the list of students who responded to a specific topic. Another view of interest to both the professor and the student is the list of students who responded to a specific question of a specific topic. On the other hand only the professor will be interested in the list of all topics that a specific student has responded to.

**Step 10:** Select the appropriate Lotus Notes templates and map these data items on the generic variables of the template.

Lotus Notes provides a set of application templates to expedite the applications development process. Our requirements gen-

eration process was devised to develop information in a form that can be mapped onto a generic application template. This mapping of the requirements onto the application template involves:

1. understanding the application template
2. mapping the data items of the application on the corresponding data items in the template
3. specifying the access rights
4. creating specific views

## ANALYSIS OF THE DISCUSSION DATABASE TEMPLATE

Using the Discussion database template, this section illustrates an approach for understanding an application template. To use this template or any other Notes template, you need to understand the architecture of the template. This involves understanding:

1. the purpose of the template
2. the structure of the documents database
3. the purpose and definition of the data fields in each form
4. the built-in views

### Purpose of the Discussion database template

The purpose of the discussion template is to provide the generic forms and views for building a discussion database application. The characteristic feature of a discussion database application is the need for a group of certified users to carry a discussion on a set of interrelated topics.

### Structure of the Discussion database template

The discussion database template is composed of a main form and a response form. The main form is used for describing the specifics of the discussion topic. The response form is used for recording the actual discussion that takes place on a discussion topic. Therefore, a discussion database application must be modeled in terms of a set of documents that contain information on the discussion topics and the a set of documents that contain the discussion on each topic.

### Purpose and definition of the data fields in the main form

The main form of the discussion database template, shown in Figure 4, contains five data fields: Subject, From, Date, Categories, and Body. Subject is a place holder for the name of the topic. Therefore, it is associated with the Topic label on the form. It is an editable text field and the user enters the topic name when the document is created. From is a place holder for the author name

and it is associated with the Author label on the form. It is a non-editable field whose data type is Document-Author. This data type automatically picks the user name from the ID when this document is created. Date is a place holder for the date and it is associated with the Date label on the form. It is a date / time field which gets the date and time from the system clock when the document is first created. Categories is associated with the Category label on the form. It is an editable keyword field which allows the user to select from a list of the categories in which the topic can be cataloged. Body is a place holder for describing the details for conducting a discussion on that topic. It is an editable rich text field.

Figure 4. MAIN TOPIC FORM

**Purpose and definition of the data fields in the response form**

The response form, shown in Figure 5, contains six fields: From, Date, DisplaySubject, Subject, Body, and OriginalSubject. The From, Date, and Body fields in the response form are similar to the From, Date, and Body data fields in the main form. DisplaySubject, OriginalSubject and Subject fields reflect the design decisions of the template designer and they need to be understood by the user.

The Subject field automatically inherits the value of the Subject data field from the main form. The user, however, can revise it. The DisplaySubject field is invisible at the time of document creation but it gets the value of the Subject data field when this document is displayed. The Subject data field is hidden when the document is displayed. This tricky linkage changes the input format of the response form document into a display format that is more meaningful and user friendly than the input format.

The OriginalSubject is a hidden field that inherits the value of the subject field in the

Figure 5. RESPONSE FORM

main form, and its value is displayed as a window label when this form is composed or read.

**The built-in views of the Discussion database template**

The Discussion Template provides three built in views. They are called the Main View, By-Author View, and Categorized View.

The Main View is a tabular summary of all the documents in the database. It displays the topics in the main form documents database and the number of discussion documents associated with that topic in the response form documents. The first column is the sequential entry number of the document, the second column contains the data document creation, the third column contains the number of responses, and the fourth column shows the name of the author who created it.

The By-Author View displays the documents created by each author irrespective of whether the document is a main document or a response document. The first column contains a number indicating the sequential position of the author in the view, the second column contains the name of the author of the document, the third column displays the document number, the fourth column shows the date the document was composed and the fifth column displays the subject of the document.

The Categorized View displays the database documents organized in terms of the keywords stored in the categories field of main documents. The first column displays the category name. Within this category, the date when the document was composed, the number of responses, and the subject and author of the main document are shown.

The above process can be used to analyze the architecture of any template. The next step in the application development process

is to clothe the template with the specifics of the application requirements. This is illustrated in the next section.

**APPLICATION IMPLEMENTATION**

Implementing the application requirements involves: Mapping the applications data items to the data elements in the template forms, Updating of template views, and Assigning access rights.

**Mapping of Applications Data Items**

The first step in the implementation of the application requirement is to associate the applications data items with the data

Main Form			
Application Data Item	Template	Data Element	
ProfessorName	From		
DateComposed	Date		
Topic	Subject		
TopicCategories	Categories		
TopicQuestions	Body		
Response Form			
Application Data Item	Template	Data Element	
StudentName	From		
TodaysDate	Date		
OriginalTopic	Subject,	OriginalSubject	
NewTopic	DisplaySubject		
AnswerDetails	Body		

elements in the template forms. The data items association between the professor-student application and the discussion database template are shown below.

**Mapping of Views**

The built-in views (i.e., the Main View, the By-Author View, and the Categories View) in the template are useful and need to be maintained in this application. They may not, however, meet all the needs of the application. For example, the professor is interested in finding out how many questions did each student answer on a particular topic. In addition, the professor wants to periodically review which topics are unanswered.

Figure 6. RESPONSE-BY-TOPIC

STUDENT	#OF RESPONSES	TOPIC
Name 1	4	Groupware
Name 2	3	
Name 3	3	
Name 1	4	Workflow
Name 3	2	

Figure 7. UNANSWERED TOPIC

DATE CREATED	TOPIC
06/10/94	Form Design
06/12/94	View Design
06/30/94	Security
06/30/94	Replication

These inquiries cannot be met with the three built-in views. New views are needed to generate answers to these questions. They are called the Response-By-Topic View and Unanswered-Topics View and are shown in figures 6 and 7 respectively. Systems for generating these extra views need to be written.

### SUMMARY

Groupware tools have the potential to do for office processes what automation did in the manufacturing process. Lotus Notes is the leading groupware tool. This paper provided an overview of the Lotus development environment, introduced an approach for analyzing Lotus Notes templates, and presented a systematic method for analyzing and documenting groupware applications. It used on a simple example of collaborative work in academic environment to illustrate the methodology.

### REFERENCES

1. Bullen, C.V. & Bennet, J.L., "Learning from User Experiences and Groupware", Proceedings of Conference on Computer-Supported Cooperative Work, Oct. 1990.
2. Chalstrom, B., "Lotus Notes, Release 3 Offers Something Good for Everyone", *InfoWorld*, Vol. 15, No. 44, Nov. 1993.
3. Ellis, C. & Rein, G., "Groupware: The Research and Development Issues", *Comm. ACM*, 1991.
4. Lotus Corporation Publication, "A System for Managing Organizational Knowledge", A Technical White Paper, Part No. 13169, April 1993, 24 pages.
5. Lotus Corporation Publication, "Lotus Workflow Strategy", A Technical White Paper, Part No. 13166, April 1993, 14 pages.
6. Lotus Corporation Publication, "Notes @ Work: Discussions", 1991, 34 pages.
7. Marca, David A., "Specifying Group Work Applications" *Proceedings of 5th International Workshop on Software Specification and Design*, May 1989.
8. Marca, David A., "Augmenting SADT™ To Develop Computer Support for Cooperative Work", *Proceedings of the IEEE Conference on Software Engineering*, May 1991.
9. Rayl, E., "Developing Groupware Apps with Lotus Notes 3.0", *Data Based Advisor*, Vol. 11, No. 5, May 1993.
10. Winograd, T., "A Language/Action Perspective on the Design of Cooperative Work" *Journal of Human-Computer Interaction*, Vol. 3; 1987.

#### Author's Biography

Donald Chand is a professor of Computer Information Systems at Bentley College. His teaching interests are in systems analysis and systems design, and his research interests are in software development processes and technologies. He has published articles in the *Communications of the ACM*, the *Journal of ACM*, and the *IEEE Software*. He has been an ACM lecturer, associate editor of the *Journal of Information Technology Management*, and a member of the visiting team for computer science accreditation. He received his Ph.D. degree from Boston University.

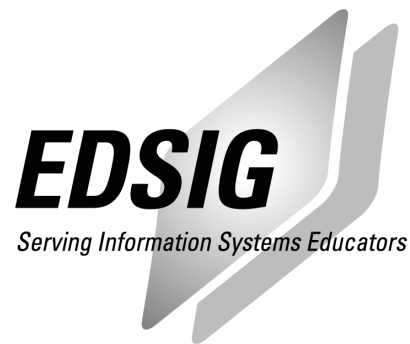
## Attend EDSIG's Annual Conference!



**EDSIG**

Serving Information Systems Educators

JUNE 11-14, 1995,  
suburban Denver, CO  
For more information,  
call 1-800-488-9204.



### **STATEMENT OF PEER REVIEW INTEGRITY**

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

Copyright ©1994 by the Information Systems & Computing Academic Professionals, Inc. (ISCAP). Permission to make digital or hard copies of all or part of this journal for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial use. All copies must bear this notice and full citation. Permission from the Editor is required to post to servers, redistribute to lists, or utilize in a for-profit or commercial use. Permission requests should be sent to the Editor-in-Chief, Journal of Information Systems Education, [editor@jise.org](mailto:editor@jise.org).

ISSN 1055-3096